Received: April 8, 2024. Revised: June 22, 2024.

International Journal of
Intelligent Engineering & Systems

http://www.inass.org/

Hybrid PSO-Adam Optimizer Approach for Optimizing Loss Function Reduction in the Dist-YOLOv3 Algorithm

Alam Rahmatulloh[1] **Ghatan Fauzi Nugraha**[1] **Irfan Darmawan**[2]*

[1]*Department of Informatics, Faculty of Engineering, Siliwangi University, Indonesia*
[2]*Department of Information Systems, Faculty of Industrial Engineering, Siliwangi University, Bandung, Indonesia*
* Corresponding author's Email: irfandarmawan@telkomuniversity.ac.id

Abstract: The rapid evolution of object recognition and classification technology is propelled by advanced supporting architectures, notably the widely embraced You Only Look Once (YOLO) architecture. YOLO is a cornerstone in this technological domain, renowned for its user-friendly implementation and a harmonious balance between accuracy and detection speed. Dist-YOLOv3, an iteration, excels in object detection and estimates object distances from the camera. In preceding research, the optimization of algorithmic performance hinged on the Adam optimizer method, yet the challenge lies in selecting the optimal initial learning rate—a pivotal factor. Adding complexity and dataset characteristics further complicate the process. Particle Swarm Optimization (PSO) emerges as a solution, automatically determining the optimal initial learning rate. In this study, we introduce an approach that combines two optimizations simultaneously, namely PSO and Adam Optimizer. Using this method can enhance the performance of the Dist-YOLOv3 model, particularly in reducing loss values. Based on the tests conducted, our developed method has proven to decrease the loss values, with a reduction of 16.2% in train loss and 12.7% in validation loss. Despite the temporal intensiveness, future research endeavors aspire to mitigate these time-related challenges. The findings underscore the effectiveness of PSO in automating the intricate process of learning rate selection and advancing model performance.

Keywords: Adam optimizer, Dist-YOLOv3, Learning rate, Particle swarm optimizer.

1. Introduction

Image recognition and classification technology, both in real-time and non-real-time scenarios, has reached high success. Architectures such as Convolutional Neural Networks (CNN) and You Only Look Once (YOLO) have become favorites due to their success in image classification challenges [1,2]. Both of these architectures can deliver a balanced outcome concerning accuracy and detection speed, particularly in the case of the YOLO architecture [2]. Currently, YOLO has reached version 8, and there are numerous variations of these versions. One of the variations is the Dist-YOLOv3 algorithm [3]. In addition to classifying objects, the algorithm can predict the distance of the detected objects from the camera. [3-5]. The utilization of YOLOv3 in the Dist-YOLOv3 algorithm is based on the speed of the detection process, the speed of bounding box determination, and the use of a lightweight architecture [3, 6-8].

To improve Dist-YOLOv3's performance, Vajgl et al. [3] optimized it using the Adaptive Moment Estimation (Adam) optimizer. However, the challenge lies in selecting the initial learning rate. The difference in choosing the Adam optimizer's learning rate can impact model training's final results [9]. Moreover, the learning rate value selection should be tailored to the dataset's characteristics. This allows for obtaining the most suitable learning rate to support the creation of the most optimal model.

Many approaches can be employed to address the challenge of selecting appropriate parameters for a method. One such method is Particle Swarm Optimization (PSO). PSO has been proven to optimize the selection of parameters effectively, resulting in well-performing models [10]. Additionally, PSO can be applied to selecting the basic CNN architecture that best suits various dataset characteristics, enhancing the model's adaptability [11,12].

There are two models produced by Vajgl et al. [3]: the class-agnostic distance YOLO (Dist-YOLOv3 G) and the class-aware distance YOLO (Dist-YOLOv3 W). Both models were trained with the same hyperparameter specifications, including an image resolution of 608x192 pixels, a learning rate for Adam set at 1e-3, a batch size of 24, and 100 epochs. The model evaluation utilized the mean Average Precision (mAP) method and Intersection over Union (IoU) as its threshold value. Dist-YOLOv3 G exhibited mAP@0.5 = 76.2% and mAP@0.95 = 30.7%, while Dist-YOLOv3 W yielded mAP@0.5 = 77.1% and mAP@0.95 = 33.5%.

This research aims to optimize the selection of learning rate and decay parameter values in the Adam optimizer, aiming to enhance the performance of the Dist-YOLOv3 model under lighter training specifications compared to previous studies. This method operates by relying on the process of refining each parameter value initialized across many swarm particles. This method makes the learning rate selection more adaptable to the utilized dataset conditions. It is anticipated that achieving this goal will result in a model of superior quality compared to the original. Furthermore, the hope is that the optimized model can be implemented across various domains related to object detection and distance prediction, such as automotive, assistive devices for disabilities, and others.

This paper consists of several main sections, namely the introduction (first section), related works (second section), proposed method (third section), and results and discussion (fourth section). Lastly, the conclusion (fifth section) summarizes the overall research conducted.

2. Related works

Many current studies utilize PSO as an alternative for optimization, especially in parameter selection. This section will elaborate on several studies employing PSO to optimize existing models.

Wang et al.'s research [12] introduces a novel method for designing Convolutional Neural Network (CNN) architectures using the Particle Swarm Optimization (PSO) algorithm, known as EPSOCNN. EPSOCNN demonstrates significant advantages in terms of computational efficiency, with an architecture design time of less than four days on GPU and a model size of 6.74M, much smaller than DenseNet's 27.2M parameters. Moreover, EPSOCNN achieves excellent accuracy performance with an error rate of 3.58% on CIFAR-10, 18.56% on CIFAR-100, and 1.84% on SVHN, and it shows good transfer learning capabilities. However, this research has limitations regarding algorithm complexity and evaluation, as it is limited to medium-scale datasets and conventional data augmentation strategies.

Fregoso et al. [13] propose two Particle Swarm Optimization (PSO)-based optimization approaches on Convolutional Neural Network (CNN) architectures, namely PSO-CNN-I and PSO-CNN-II, for sign language letter recognition using three databases: ASL alphabet, ASL MNIST, and MSL alphabet. Experimental results show that the PSO-CNN-I approach yields the best recognition rates, averaging 99.58% on the ASL alphabet and 99.53% on ASL MNIST. In comparison, PSO-CNN-II excels on the MSL database with an average of 98.91%. The main advantage of this research is the PSO approach's ability to find optimal CNN architectures that minimize parameters and maximize recognition rates. However, this study has several drawbacks, including high computational complexity and the need for intensive parameter adjustments.

The research paper by [14] presents a PSO-CNN-based deep learning model for predicting forest fire risk on a national scale, achieving an accuracy of 82.2% and an AUC value of 0.92, outperforming other models like logistic regression, random forest, support vector machine, k-nearest neighbors, and traditional CNN models. The model's stability and consistent results across multiple runs demonstrate its potential for stable predictions, especially in high-risk scenarios, aiding in efficient resource allocation for fire prevention and management. However, the study did not consider spatial heterogeneity in fire locations, suggesting future research could integrate this aspect with deep learning to enhance prediction models. Overall, the PSO-CNN model showcases superior performance in forest fire risk prediction, offering a promising direction for future research in deep learning for ecological disaster management.

Elhani et al. [15] presents compelling results with improved error rates across various datasets. For instance, on the MNIST dataset, the proposed method achieves the best error rate 0.30 compared to 0.32 with PSOCNN and 0.35 with PSO-based approaches. Similarly, on datasets like MNIST-BI, MNIST-RD, and MNIST-RDBI, the proposed method consistently outperforms competitors, showcasing 2.27, 2.74, and 11.47 error rates, respectively. These results highlight the efficiency and practicality of the approach in optimizing Convolutional Neural Network (CNN) architectures for image classification tasks. However, limitations include potential sensitivity to initialization and the need for further exploration of diverse datasets.

The study by [16] demonstrates that the enhanced PSO variants effectively identify relevant features, improving object recognition performance. By optimizing the feature selection process, the model achieves better accuracy and robustness in recognizing objects from input data. The research showcases a substantial enhancement in network throughput by 25% and a notable reduction in latency by 15% compared to conventional resource allocation methods. These improvements highlight the efficacy of the proposed machine learning-based approach in optimizing resource allocation dynamically in wireless networks. Despite its strengths, challenges such as the computational complexity of implementing machine learning models, the need for substantial training data, and the requirement for continuous model updates to accommodate evolving network dynamics may present hurdles to practical deployment. These factors could limit the proposed approach's scalability and ease of adoption in real-world network settings.

An extensive literature review reveals that despite significant advancements in the utilization of algorithm-based optimization, such as Particle Swarm Optimization (PSO), there are still opportunities for improvement that can be explored. These opportunities include addressing computational complexity, enhancing the generalization and stability of methods, and integrating with emerging techniques. This research offers a low computational complexity optimization method for the Dist-YOLOv3 algorithm using the PSO-Adam Optimizer.

3. Proposed method

The research methodology employed adapts from the design framework process developed by John et al. [17] in developing Machine Learning or Deep Learning models. Consequently, the research methodology is simplified, as depicted in Fig. 1.

The research begins with a literature review focusing on similar studies related to object recognition models and object distance estimation predictions. The selected articles are sourced from accredited journals, ensuring high-quality references for this research. Additionally, the literature revie generates a comprehensive overview that forms a solid foundation for the study.

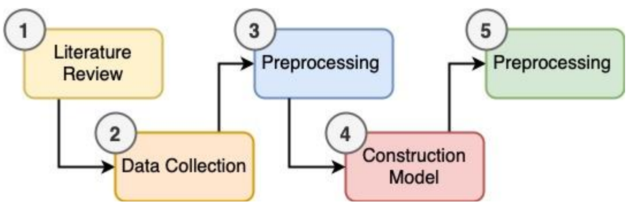

Figure. 1 Research method

Next, the data collection process involves searching for datasets widely used by academics, which serve as benchmarks for creating object recognition models and object distance estimation predictions.

This is done to facilitate comparisons with other studies using the same dataset, allowing existing comparisons to represent the strengths and weaknesses of the model resulting from this research against the benchmark dataset. Once the data collection process is complete, the next step involves preprocessing the obtained dataset. This stage includes annotation of images, similar to the approach taken by Vajgl et al. [3], by initializing bounding boxes and the actual object distance to the camera. The output of this process will generate a file containing numerical representations of the bounding box values and actual object distance values.

The core of this research lies in the model creation phase. Model creation is executed using the flow and algorithm proposed by Vajgl et al. [3], namely the Dist-YOLOv3 algorithm. Despite the satisfactory results produced by the model based on this algorithm, there remains a possibility for further optimization. Vajgl et al. [3] used the Adam optimizer algorithm to enhance the performance of the Dist-YOLOv3 algorithm. However, a challenge

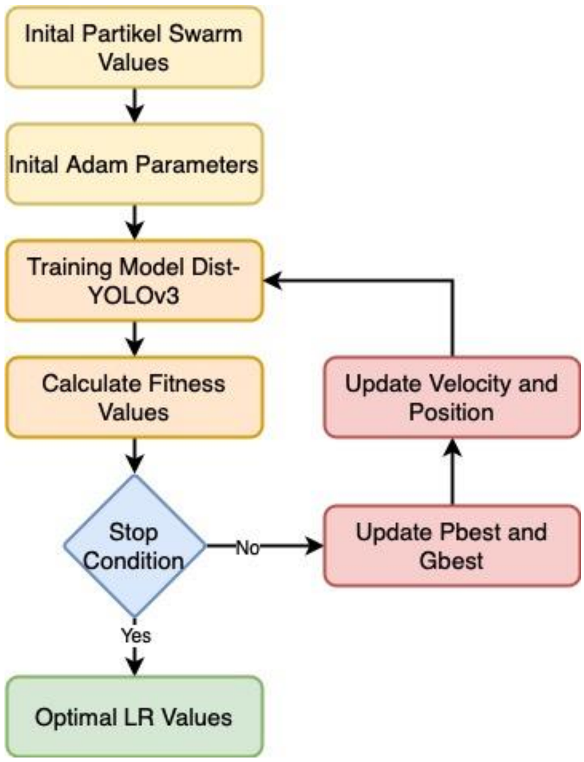

Figure. 2 Proposed method

arises in manually estimating the learning rate initialization value, even though the study incorporates Cosine Annealing for periodic learning rate reduction. As an improvement, Particle Swarm Optimization (PSO) is proposed as a solution for precise learning rate initialization. Using inertia weight in PSO has enhanced its performance, making it suitable for implementation in this model creation [18,19].

Fig. 2 illustrates the proposed method for enhancing the Dist-YOLOv3 algorithm by adjusting the initialization of parameters in the Adam optimizer using PSO. This approach is an extension of the method utilized by Li et al. [20]. The initialization process for the particle swarm is conducted randomly, adhering to manually specified parameter values, including the determination of initial velocity values [21]. Manual determination of the Adam optimizer parameters is initially performed to facilitate the training process of the algorithmic model. The fitness value is computed by evaluating the train loss and validation loss values in the Dist-YOLOv3 model using the approach described by Vajgl et al. [3].

$$l_{train} = \sum_{i=0}^{G^w G^h} \sum_{j=0}^{n^a} q_{i,j}[l_1(i,j) + l_2(i,j) + l_3(i,j) + l_5(i,j)] + l_4(i,j) \quad (1)$$

$$l_{val} = \sum_{i=0}^{G^w G^h} \sum_{j=0}^{n^a} q_{i,j}[l_1(i,j) + l_2(i,j) + l_3(i,j) + l_5(i,j)] + l_4(i,j) \quad (2)$$

The variables l_{train} and l_{val} denote the train loss and validation loss, respectively. They are computed based on $l_1(i,j)$ for bounding box prediction loss, $l_2(i,j)$ for box dimension loss, $l_3(i,j)$ for confidence loss, $l_4(i,j)$ for class prediction loss, and $l_5(i,j)$ for distance estimation prediction loss. The PSO formula implemented includes the inertia weight multiplier ($w(t)$).

$$v_i(t+1) = w(t)v_i + c_1 r_1\left(pbest_i - x_i(t)\right) + c_2 r_2(gbest_i - x_i(t)) \quad (3)$$

$$x_i(t+1) = x_i(t) + v_i(t+1) \quad (4)$$

Where the value $v_i(t+1)$ represents the particle velocity at iteration t+1 obtained from the calculation involving $w(t)$ as the inertia weight at iteration t, nilai v_i is the particle velocity at iteration t, c_1, and c_2 are the acceleration constants or cognitive constants, r_1 and r_2 are random values within the range of 0 to 1, $pbest_i$ is the best position achieved by particle i, $gbest_i$ is the best position ever achieved by the entire population, and $x_i(t)$ is the value of the position of particle i at iteration t. The value $x_i(t+1)$ is the updated position of the particle at iteration t+1, obtained by adding the value of $x_i(t)$ to $v_i(t+1)$.

The formula in the Adam optimizer that the updates performed by PSO will influence is the part related to weight updates in the hidden layer used in the architecture of Dist-YOLOv3.

$$\omega_t = \omega_{t-1} - \alpha \frac{m'_t}{\sqrt{v'_t + \varepsilon}} \quad (5)$$

Where the value $\alpha = gbest$ represents the learning rate that will be continuously updated to obtain the most optimal α, the utilization of formulas (1)-(5) will persist if the stop condition in the proposed method has not been reached. The stop condition employed involves the number of iterations used to search for the optimal $\alpha = gbest$ using PSO. Besides searching for the most optimal α value, the value ε will be sought to assess its effectiveness in reducing the loss function. Therefore, formula (5) can be modified into the formula (6).

$$\omega_t = \omega_{t-1} - gbest[0] \frac{m'_t}{\sqrt{v'_t + gbest[1]}} \quad (6)$$

Formula (6) will be utilized to update the weight values in the hidden layer used in Dist-YOLOv3. Additionally, periodic learning rate reduction is performed using Cosine Annealing, as implemented by Vajgl et al. [3].

This research's final stage is to evaluate using the Average Precision (AP) or mean Average Precision (mAP) metric. This method is commonly employed to measure the performance of object recognition models [22]. The mAP metric involves calculating precision-recall metrics and determining the accuracy of positive predictions using Intersection over Union (IoU).

4. Result and discussion

The dataset used in the conducted experiments is the KITTI (Karlsruhe Institute of Technology and Toyota Technological Institute) 3D Object Detection Evaluation 2017 [23]. Academics commonly employ this dataset as a benchmark for evaluating object recognition models [3]. The KITTI 3D Object Detection Evaluation 2017 consists of 7481 training data and 7518 testing data, but only the training data will be utilized. This decision is based on the fact that the available training data comes with actual distance

Received: April 8, 2024. Revised: June 22, 2024.

labels, aiding the model in predicting distances in detected images. Out of the 7481 data points, they will be divided into three subsets: 6030 training data, 782 test data, and 669 validation data.

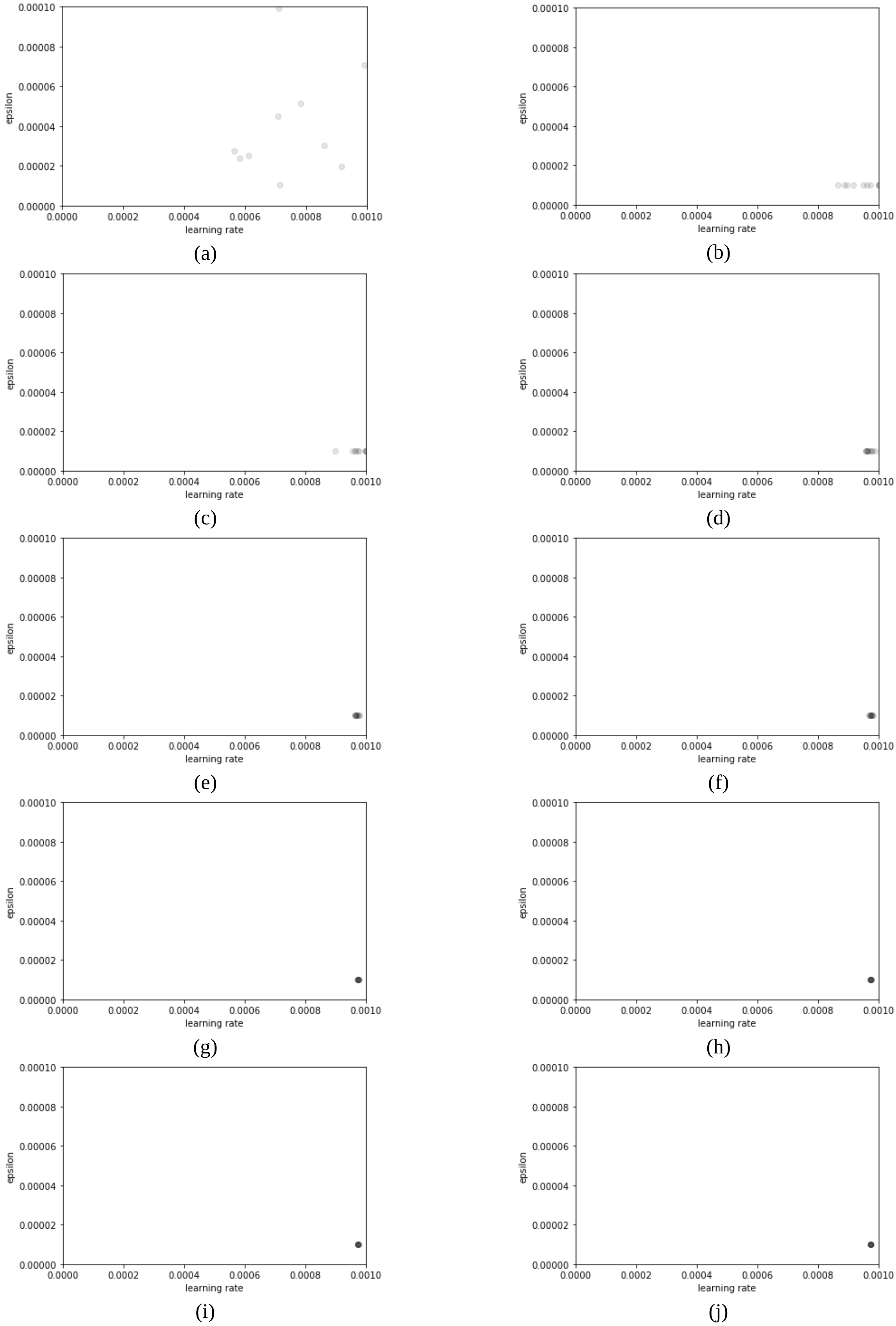

Figure. 3 The obtained results for the search of α and ε values using PSO: (a)Iteration 1, (b)Iteration 2, (c)Iteration 3, (d)Iteration 4, (e)Iteration 5, (f)Iteration 6, (g)Iteration 7, (h)Iteration 8, (i)Iteration 9, and (j)Iteration 10

Received: April 8, 2024. Revised: June 22, 2024.

Table 1. Initialization of α and ε randomly

Particle to	α	ε
1	7.09941485e-04	4.50058156e-05
2	8.60291591e-04	3.01319574e-05
3	7.15760274e-04	1.04433448e-05
4	9.91611617e-04	7.09323456e-05
5	5.66228971e-04	2.74740948e-05
6	5.84657350e-04	2.36932942e-05
7	9.16550427e-04	1.95675586e-05
8	6.14126924e-04	2.50106011e-05
9	7.82396667e-04	5.16069723e-05
10	7.12620725e-04	9.91002519e-05

Table 2. The final values for α and ε

Particle to	α	ε
1	9.76028217e-04	1.00000000e-05
2	9.75627030e-04	1.00000000e-05
3	9.76058087e-04	1.00000000e-05
4	9.76013563e-04	1.00000000e-05
5	9.76036401e-04	1.00000000e-05
6	9.76037595e-04	1.00000000e-05
7	9.76945545e-04	1.00000000e-05
8	9.75967929e-04	1.00000000e-05
9	9.76028213e-04	1.00000000e-05
10	9.76027853e-04	1.00000000e-05

In the pursuit of finding the most optimal values for α and ε, the initialization of PSO variables is performed with c1 = c2 = 2, ωmax = 0.9, ωmin = 0.3, 10 iterations, and a total of 10 particles. Additionally, training model specifications are required to seek the values of train loss and validation loss as metrics for the PSO fitness function. These specifications include an image input size of 160x160, two epochs, and a batch size 24. To enhance model optimization, the Darknet53 architecture in YOLOv3 is replaced with the lighter-weight Xception architecture, which utilizes fewer parameters [24]. The results of the parameter search for α and ε in the Adam optimizer can be observed in Fig. 3.

Fig. 3 illustrates the movement of the particle swarm from iteration 1 to iteration 10. The graph at iteration 1 shows the particle swarm as the initial values for α and ε. Table 1 displays the randomly determined initial values for α and ε.

The values in Table 1 will continuously be updated, considering the values of $pbest_i$ and $gbest_i$. $pbest_i$ is updated if the train loss and validation loss values in the next iteration are smaller, but it must still consider the best values for all particles or the value of $gbest_i$. The value of $gbest_i$ will influence the position of each particle when approaching the value of $gbest_i$. As a result, all particles will converge towards a single point, with the movement regulated by the velocity of each particle. As seen in Fig. 3, periodically, in each iteration, all particles will converge towards a point considered to be the most optimal.

By the second iteration, all ε values for each particle are compacted to the value of 1.00000000e-05 until the last iteration. This is because the search process for ε is constrained to prevent a slow-down in the early stages of training. ε significantly affects the training process as it determines the learning reduction rate and prevents zero division in the weight update process [25]. Table 2 displays the values of the final position for each particle or the concluding values for α and ε.

From the values in Table 2, one particle is selected as the global best ($gbest$), which has the values α = 9.76028217e-04 and ε = 1.00000000e-05. These two values will be tested in the comprehensive training process to compare the results with the model created by Vajgl et al. [3], available at the link https://gitlab.com/EnginCZ/yolo-with-distance. The model in this link was generated using 43 epochs, with training specifications as outlined in Vajgl et al. study [3]. The model training process in this research is conducted with 40 epochs, an image size of 160x160, and a batch size of 24.

Figure. 4 Comparison of train loss and validation loss values

Received: April 8, 2024. Revised: June 22, 2024.

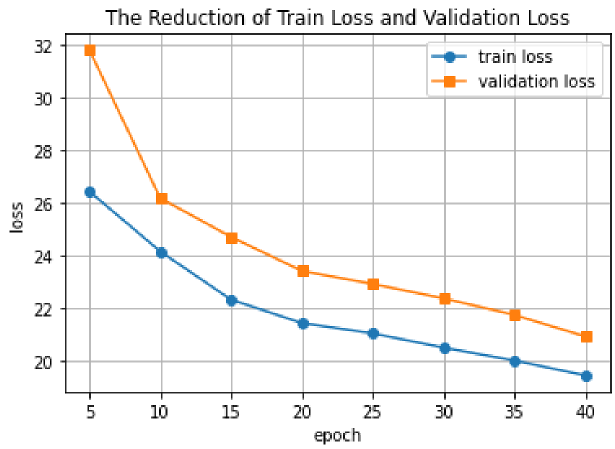

Figure. 5 The decrease in train loss and validation loss values

Fig. 4 illustrates the difference between the model by Vajgl et al. [3] and the model in this study, a result of the search for optimal values for α and ε using PSO. Properly selecting learning rate values in the Adam optimizer optimization method can significantly improve the Dist-YOLOv3 model. To further clarify the existing differences, Fig. 5 shows the decrease in loss values and validation loss in this research model.

A gap becomes apparent as the training process reaches the 15th epoch, where the train loss value has reached 22.339, indicating that this value is already below the original Dist-YOLOv3 model's value. The validation loss value is also at 23.427 as the training progresses into the 20th epoch. This demonstrates that the proper choice of learning rate can effectively reduce the loss values. The next step is to compare the models using the mAP metric method. Figs. 6 to 8 depict the differences between the two compared models in precision, recall, and mAP metric values.

The precision values obtained by this research model are below those obtained by the model from [3]. This indicates that the object detection results when the model is tested contain many false negatives. False negatives occur because the model detects objects that are not actually present but considers them to be present.

The recall values experience a significant decrease, almost half of those obtained by the model by Vajgl et al. [3]. This decline is attributed to this research model's tendency to miss many objects that should have been detected.

The same pattern is observed in the mAP measurement, indicating a significant decrease compared to the model by Vajgl et al. [3]. Small precision and recall values will result in a small mAP value. Evaluation using the mAP method shows that

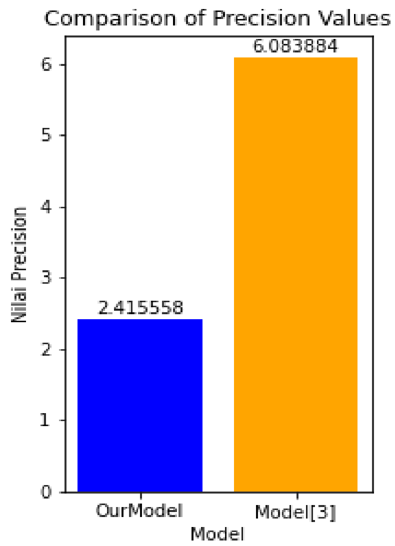

Figure. 6 Comparison of precision values

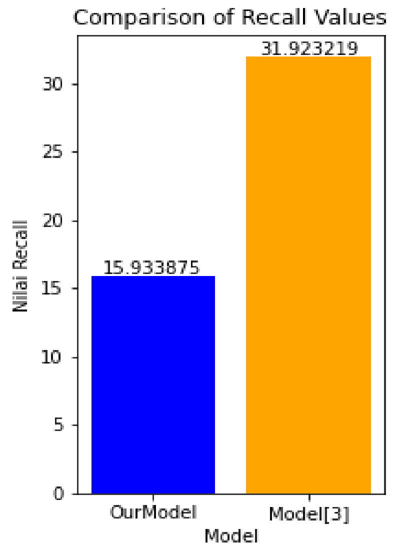

Figure. 7 Comparison of Recall Values

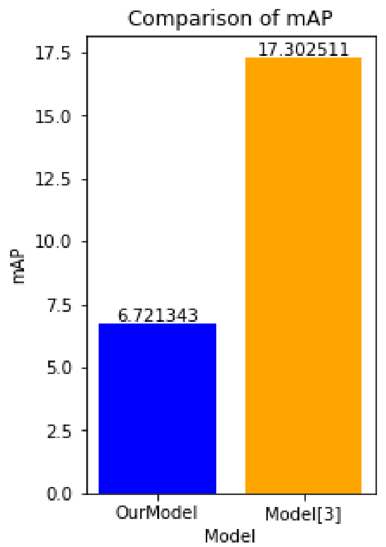

Figure. 8 Comparison of mAP

Received: April 8, 2024. Revised: June 22, 2024.

Table 3. Comparison of error rates for the MNIST dataset

Research	Model	Error Rate (mean)
Miao et al. [27]	sosCNN	0.38%
Lawrence et al. [28]	PSO-Based Model	0.38%
Nistor & Czibula [29]	IntelliSwAS	0.38%
Elhani et al. [15]	pswvCNN	0.44%
Our Method	Hybrid PSO-Adam Optimizer	0.38%

Table 4. Comparison of error rates for the Fashion-MNIST dataset

Research	Model	Error Rate (mean)	parameter	epoch
Miao et al. [27]	sosCNN	6.17%	3.42 M	100
Elhani et al. [15]	pswvCNN	6.1%	3.74 M	100
Our Method	Hybrid PSO-Adam Optimizer	7.48%	242 K	100

the precision, recall, and mAP values of this research model still cannot surpass those of the model by Vajgl et al. [3]. The main factor that makes the model by Vajgl et al. [3] superior is the larger input image size, specifically using images with dimensions of 608x608. This aspect contributes to the original Dist-YOLOv3 model performing better when evaluated using the mAP method. This experiment demonstrates that achieving a small loss value sometimes translates into a better model evaluation.

To further demonstrate the effectiveness of our proposed method, we conducted tests on the CNN architecture using various types of datasets. The results of these tests will be compared with those of related studies and other research focusing on CNN architecture development. Table 3 compares error rates obtained for the CNN model using the MNIST dataset [26].

Table 3 shows that by adjusting the learning rate and epsilon values in the Adam Optimizer parameters, the CNN model can perform exceptionally well, even matching the performance of CNNs with modified architectures, with an error rate of 0.38%. In the studies listed in Table 3, PSO was employed to select the best architecture, thus increasing the complexity of the CNN architecture. The following comparison involves using the Fashion-MNIST dataset [30], as shown in Table 4.

Table 5. Comparison of Mean Accuracy for ASL-MNIST and ASL-Alphabet Datasets

Reference	Model	ASL-MNIST	ASL-Alphabet
Fregoso et al. [13]	PSO-CNN-I	99.58%	99.53 %
Fregoso et al. [13]	PSO-CNN-II	99.48%	98.69 %
Rodriguez et al. [31]	Improved CNN	97.64 %	-
Ma et al. [32]	TSM-ResNet50	99.09 %	97.57 %
Mannan et al. [33]	DeepCNN	99.67 %	-
Our Method	Hybrid PSO-Adam Optimizer	99.91%	99.68%

Table 6. Comparison of mean accuracy for CNN on the UCI-HAR dataset

Reference	Model	UCI-HAR
Ankalaki & Thippeswamy [35]	OPTCovNet	99.72%
Sikder et al. [36]	Two Channel CNN	95.25%
Phukkan et al. [37]	6-Layer CNN	91.18 %
Ismail et al. [38]	AUTO-HAR	98.5 %
Our Method	Hybrid PSO-Adam Optimizer	99.97%

The comparison results in Table 4 indicate that our study's error rate is the highest. This suggests that the performance quality is still below that of other methods. However, our model uses only 242K parameters, making it lighter than other models. The next comparison uses the ASL-MNIST and ASL-Alphabet datasets, as shown in Table 5.

Using the Hybrid PSO-Adam Optimizer method on the CNN architecture for the ASL dataset demonstrates good performance, achieving the highest mean accuracy. The accuracy for the ASL-MNIST dataset is 99.91%, while for the ASL-Alphabet dataset, it is 99.68%. Finally, our developed method will be compared with CNN models using the Human Activity Recognition dataset, specifically UCI-HAR [34].

The results of the comparison in Table 6 further demonstrate the quality of our proposed method. It shows the highest accuracy among other studies, with an accuracy score of 99.97% on the UCI-HAR dataset. This proves that using the Hybrid PSO-Adam Optimizer on CNN architecture can enhance the architecture's performance across various datasets, including image data and time series data.

Received: April 8, 2024. Revised: June 22, 2024.

5. Conclusion

This study successfully demonstrates that the proper selection of the learning rate can reduce the loss values during training and validation. The use of PSO to find optimal values for α and ε in filling the hyperparameters of the Adam optimizer works well and yields the most optimal values. Among the ten randomly generated values as particles, the best global value is obtained and used as the learning rate and epsilon for the hyperparameters of the Adam optimizer. The best global values are α=9.76028217e-04 and ε=1.00000000e-05. Furthermore, the experiment in training the Dist-YOLOv3 model for 40 epochs proves that this research model has minor train loss and validation loss values compared to the original model, even though the original model was trained for 43 epochs. Next, to demonstrate that our method significantly impacts a model, we conducted tests on several benchmark datasets. The test results show an improvement in the model's performance, with scores nearly surpassing all existing studies. Future work to enhance this research includes refining the evaluation metrics using the mAP method and optimizing the parameter search time. This way, it is expected that, in addition to achieving small loss values, the model can have a more considerable mAP value and a shorter time in finding the most optimal parameter values.

Conflicts of Interest

The authors declare no conflict of interest.

Author Contributions

Alam Rahmatulloh conceived and designed the research and contributed to writing the manuscript. Ghatan Fauzi Nugraha collected data, conducted experiments, analyzed data, and drafted the initial version of the manuscript. Irfan Darmawan contributed to writing the paper and conducted data analysis. All authors have approved the final version.

Acknowledgments

Telkom University supported this work under the Institute for Research and Community Service.

References

[1] J. Plested, and T. Gedeon, "Deep transfer learning for image classification: a survey", 2022. Online]. Available: https://arxiv.org/abs/2205.09904

[2] J. Terven, D. M. Córdova-Esparza, and J. A. Romero-González, "A Comprehensive Review of YOLO Architectures in Computer Vision: From YOLOv1 to YOLOv8 and YOLO-NAS", *Mach. Learn. Knowl. Extr.*, Vol. 5, No. 4, pp. 1680-1716, 2023.

[3] M. Vajgl, P. Hurtik, and T. Nejezchleba, "Dist-YOLO: Fast Object Detection with Distance Estimation", *Appl. Sci.*, Vol. 12, No. 3, pp. 1-13, 2022.

[4] M. M. Faisal, *et al.*, "Object Detection and Distance Measurement Using AI", In: *Proc. of 2021 14th International Conference on Developments in eSystems Engineering (DeSE)*, Vol. 21, pp. 559-565.

[5] O. Bourja, H. Derrouz, H. A. Abdelali, A. Maach, R. O. H. Thami, and F. Bourzeix, "Real Time Vehicle Detection, Tracking, and Inter-vehicle Distance Estimation based on Stereovision and Deep Learning using Yolov3", *Int. J. Adv. Comput. Sci. Appl.*, Vol. 12, No. 8, pp. 915-923, 2021.

[6] K. J. Oguine, O. C. Oguine, and H. I. Bisallah, "YOLO v3: Visual and Real-Time Object Detection Model for Smart Surveillance Systems(3s)", In: *Proc. of 5th Int. Conf. Inf. Technol. Educ. Dev. Chang. Narrat. Through Build. a Secur. Soc. with Disruptive Technol. ITED 2022*, 2022.

[7] L. Zhao, and S. Li, "Object Detection Algorithm Based on Improved YOLOv3", *Electronics*, Vol. 9, No. 3, 2020.

[8] Y. Li, L. Rong, R. Li, and Y. Xu, "Fire Object Detection Algorithm Based on Improved YOLOv3-tiny", In: *Proc. of 2022 7th Int. Conf. Cloud Comput. Big Data Anal. ICCCBDA 2022*, pp. 264-269, 2022.

[9] S. Jiang, K. Guo, J. Liao, and G. Zheng, "Solving Fourier ptychographic imaging problems via neural network modeling and TensorFlow", *Biomed. Opt. Express*, Vol. 9, No. 7, p. 3306, 2018.

[10] M. O. Mustafa, "Optimal Parameter Values of Pid Controller for DC Motor Based on Modified Particle Swarm Optimization with Adaptive Inertia Weight", *Eastern-European J. Enterp. Technol.*, Vol. 1, pp. 35-45, 2021.

[11] F. E. Fernandes Jr, and G. G. Yen, "Particle swarm optimization of deep neural networks architectures for image classification", *Swarm Evol. Comput.*, Vol. 49, No. July, pp. 62-74, 2019.

[12] B. Wang, B. Xue, and M. Zhang, "Particle Swarm optimisation for Evolving Deep Neural Networks for Image Classification by Evolving and Stacking Transferable Blocks", In: *Proc. of*

2020 IEEE Congress on Evolutionary Computation (CEC), pp. 1–8, 2020.

[13] J. Fregoso, C. I. Gonzalez, and G. E. Martinez, “Optimization of convolutional neural networks architectures using pso for sign language recognition”, *Axioms*, Vol. 10, No. 3, 2021.

[14] X. You, *et al.*, “A PSO-CNN-Based Deep Learning Model for Predicting Forest Fire Risk on a National Scale”, *Forests*, Vol. 15, No. 1, 2024.

[15] D. Elhani, A. C. Megherbi, A. Zitouni, F. Dornaika, S. Sbaa, and A. Taleb-Ahmed, “Optimizing convolutional neural networks architecture using a modified particle swarm optimization for image classification”, *Expert Syst. Appl.*, Vol. 229, 2023.

[16] M. Willis, L. I. Zhang, H. A. N. Liu, H. Xie, and K. Mistry, “Object recognition using enhanced particle swarm optimization”, In: *Proc. of Int. Conf. Mach. Learn. Cybern.*, Vol. 2020-Decem, No. December, pp. 241-246, 2020.

[17] M. M. John, H. H. Olsson, and J. Bosch, “Developing ML / DL Models : A Design Framework”, In: *Proc. of 2020 IEEE/ACM International Conference on Software and System Processes (ICSSP)*, pp. 1-10, 2020.

[18] D. Cekus, and D. Skrobek, “The influence of inertia weight on the Particle Swarm Optimization algorithm”, *J. Appl. Math. Comput. Mech.*, Vol. 17, No. 4, pp. 5-11, 2018.

[19] S. Kessentini, and D. Barchiesi, “Particle Swarm Optimization with Adaptive Inertia Weight”, *Int. J. Mach. Learn. Comput.*, Vol. 5, No. 5, pp. 368-373, 2015.

[20] X. Li, S. Wu, X. Li, H. Yuan, and D. Zhao, “Particle Swarm Optimization-Support Vector Machine Model for Machinery Fault Diagnoses in High-Voltage Circuit Breakers”, *Chinese J. Mech. Eng. (English Ed.*, Vol. 33, No. 1, 2020.

[21] A. Engelbrecht, “Particle swarm optimization: Velocity initialization”, In: *Proc. of 2012 IEEE Congress on Evolutionary Computation*, Vol. 12, pp. 1-8.

[22] J. Terven, and D. Cordova-Esparza, “A Comprehensive Review of YOLO: From YOLOv1 to YOLOv8 and YOLO-NAS”, *Mach. Learn. Knowl. Extr.*, Vol. 5, No. 4, pp. 1680-1716, 2023.

[23] A. Geiger, P. Lenz, and R. Urtasun, “Are we ready for autonomous driving? The KITTI vision benchmark suite”, In *Proc. of 2012 IEEE Conference on Computer Vision and Pattern Recognition*, Vol. 12, pp. 3354-3361.

[24] F. Chollet, “Xception: Deep learning with depthwise separable convolutions”, In: *Proc. of 30th IEEE Conf. Comput. Vis. Pattern Recognition, CVPR 2017*, Vol. 2017-Janua, pp. 1800-1807, 2017.

[25] W. Yuan, and K. Gao, “EAdam Optimizer : How Impact Adam”, *arXiv : 2011. 02150v1 [cs. LG]*, Vol. 4, pp. 1-13, 2020,

[26] Y. Lecun, L. Bottou, Y. Bengio, and P. Haffner, “Gradient-based learning applied to document recognition”, In: *Proc. of IEEE*, Vol. 86, No. 11, pp. 2278-2324, 1998.

[27] F. Miao, L. Yao, and X. Zhao, “Evolving convolutional neural networks by symbiotic organisms search algorithm for image classification”, *Appl. Soft Comput.*, Vol. 109, p. 107537, 2021.

[28] T. Lawrence, L. Zhang, C. P. Lim, and E. J. Phillips, “Particle Swarm Optimization for Automatically Evolving Convolutional Neural Networks for Image Classification”, *IEEE Access*, Vol. 9, pp. 14369-14386, 2021.

[29] S. C. Nistor, and G. Czibula, “IntelliSwAS: Optimizing deep neural network architectures using a particle swarm-based approach”, *Expert Syst. Appl.*, Vol. 187, p. 115945, 2022.

[30] H. Xiao, K. Rasul, and R. Vollgraf, “Fashion-MNIST: a Novel Image Dataset for Benchmarking Machine Learning Algorithms”, *arXiv, vol. abs/1708.0*, pp. 1-6, 2017.

[31] R. Rodriguez, C. I. Gonzalez, G. E. Martinez, and P. Melin, “An Improved Convolutional Neural Network Based on a Parameter Modification of the Convolution Layer”, In: *Proc. of Fuzzy Logic Hybrid Extensions of Neural and Optimization Algorithms: Theory and Applications*, O. Castillo and P. Melin, Eds., pp. 125-147, 2021.

[32] Y. Ma, T. Xu, and K. Kim, “Two-Stream Mixed Convolutional Neural Network for American Sign Language Recognition”, *Sensors*, Vol. 22, No. 16, 2022 ,

[33] A. Mannan, A. Abbasi, A. R. Javed, A. Ahsan, T. R. Gadekallu, and Q. Xin, “Retracted: Hypertuned Deep Convolutional Neural Network for Sign Language Recognition”, *Comput. Intell. Neurosci.*, Vol. 2023, pp. 1-1, 2023.

[34] D. Anguita, A. Ghio, L. Oneto, X. Parra, and J. L. Reyes-Ortiz, “A public domain dataset for human activity recognition in free-living conditions”, In: *Proc. of - ESANN 2013 proceedings, Eur. Symp. Artif. Neural Networks, Comput. Intell. Mach. Learn.*, No. April, pp. 437-442, 2013.

[35] S. Ankalaki, and M. N. Thippeswamy, “Optimized Convolutional Neural Network

Using Hierarchical Particle Swarm Optimization for Sensor Based Human Activity Recognition", *SN Comput. Sci.*, Vol. 5, No. 5, 2024.

[36] N. Sikder, M. S. Chowdhury, A. S. M. Arif, and A. Al Nahid, "Human activity recognition using multichannel convolutional neural network", In: *Proc. of 2019 5th Int. Conf. Adv. Electr. Eng. ICAEE 2019*, pp. 560-565, 2019.

[37] N. Phukan, S. Mohine, A. Mondal, M. S. Manikandan, and R. B. Pachori, "Convolutional Neural Network-Based Human Activity Recognition for Edge Fitness and Context-Aware Health Monitoring Devices", *IEEE Sens. J.*, Vol. 22, No. 22, pp. 21816-21826, 2022.

[38] W. N. Ismail, H. A. Alsalamah, M. M. Hassan, and E. Mohamed, "AUTO-HAR: An adaptive human activity recognition framework using an automated CNN architecture design", *Heliyon*, Vol. 9, No. 2, p. e13636, 2023.